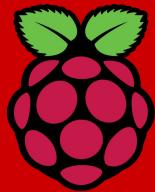
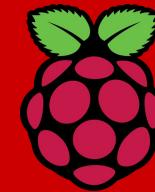


Getting Started with the Raspberry Pi: *Building and Controlling* *Beginner Projects*

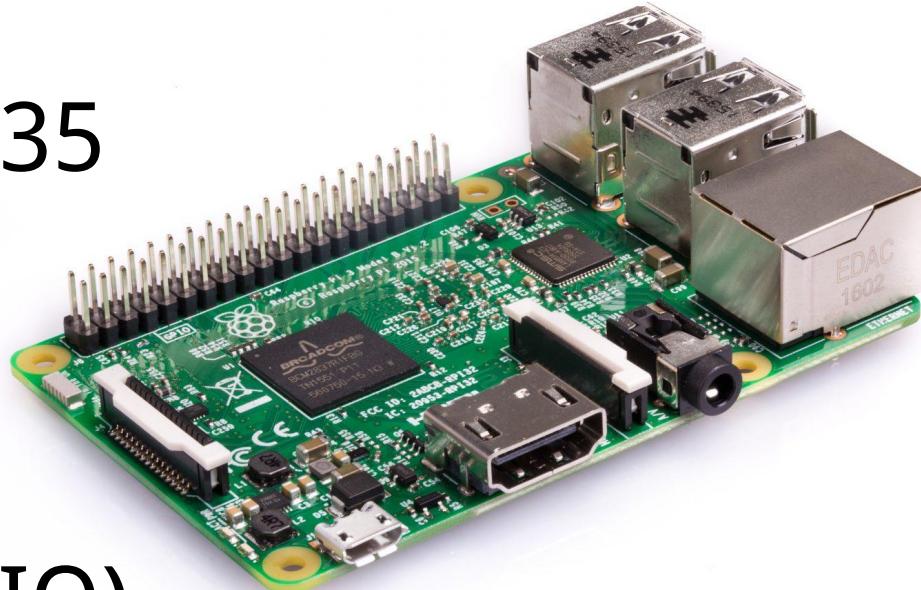


NCSU LIBRARIES



What is a Raspberry Pi?

- A full computer
- affordable: \$5-\$35
- Linux / OSS
- ecosystem + community
- DIY-friendly (GPIO)

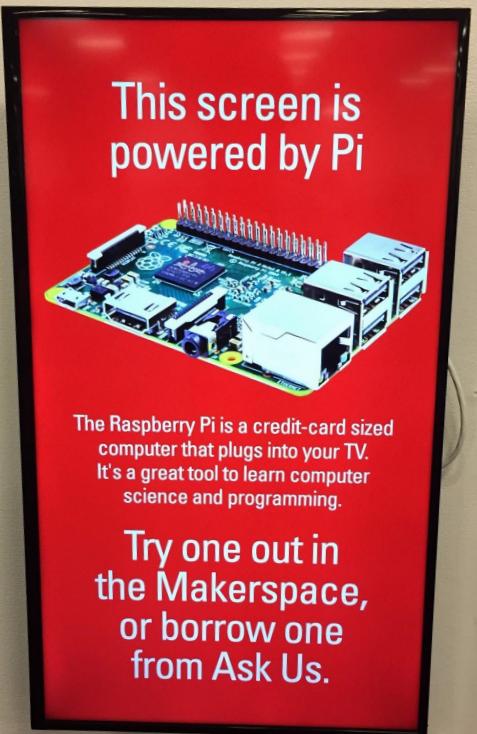


Raspberry Pi in Education



- Pi Foundation
- Teaching Computer Science, STEAM
- K-12 + Higher Ed.
- A Full Curriculum at [raspberrypi.org](https://www.raspberrypi.org)

Pi in the Libraries: Projects



- Card Access
- Poetry Bot
- LulzBots
- Door Count
- Signage



An Example NCSU Pi Project

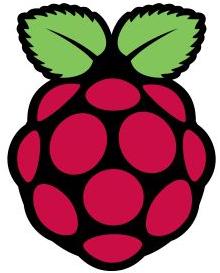
Intimate Fields

Being an exploration of Posies, silks, rings and other wearable intimacies. Touch a ring or other tagged object to the NFC reader to receive a secret note.

Follow on Twitter @intimatefields

by Helen Burgess & Margaret Simon, NCSU Dept. of English



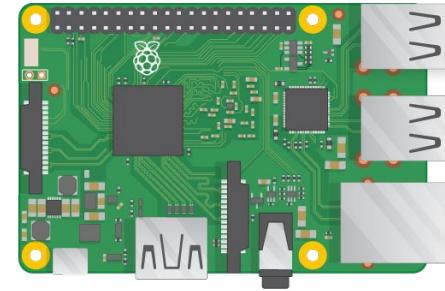
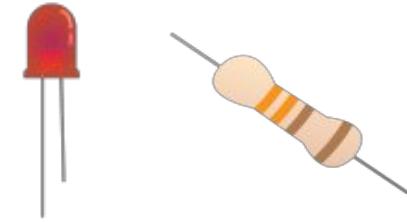


Let's Get Physical

Scratch & Python

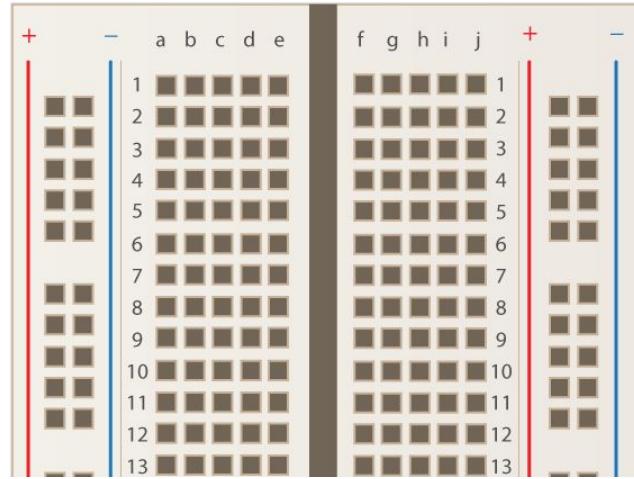
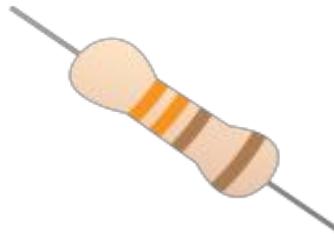
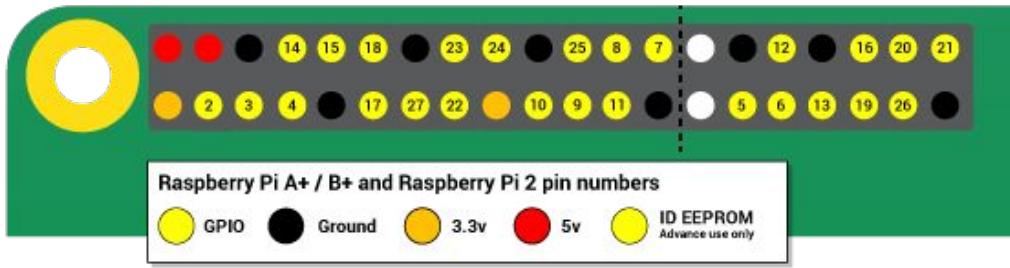
Goals

- Why physical computing?
- Build a simple circuit using Raspberry Pi
- Control an output using Scratch
- Create circuits & programs with Python



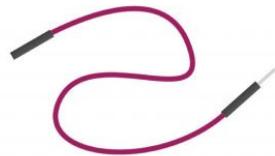
Physical Computing

Enables creators to connect the virtual world with the physical, great for engagement.

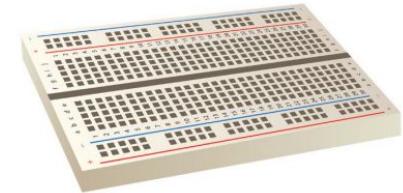
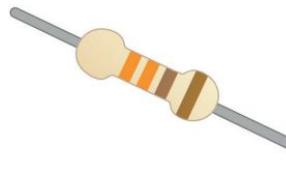


Materials

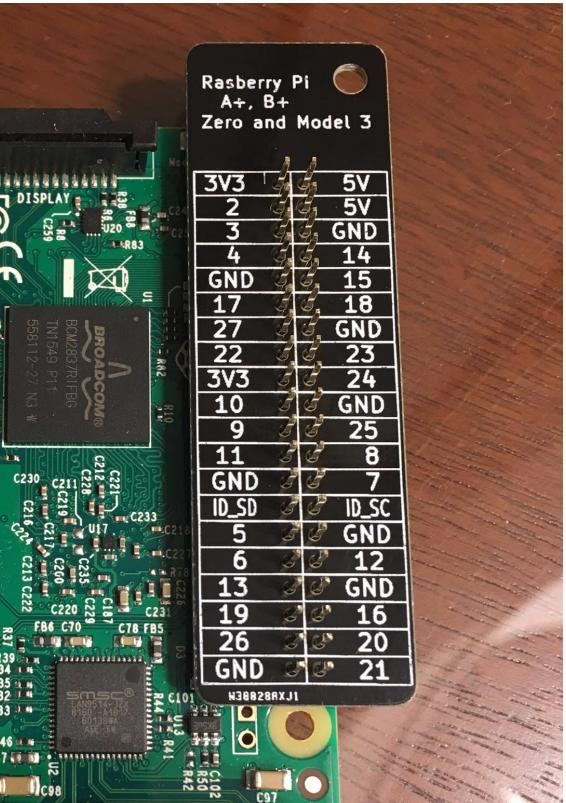
You will need:



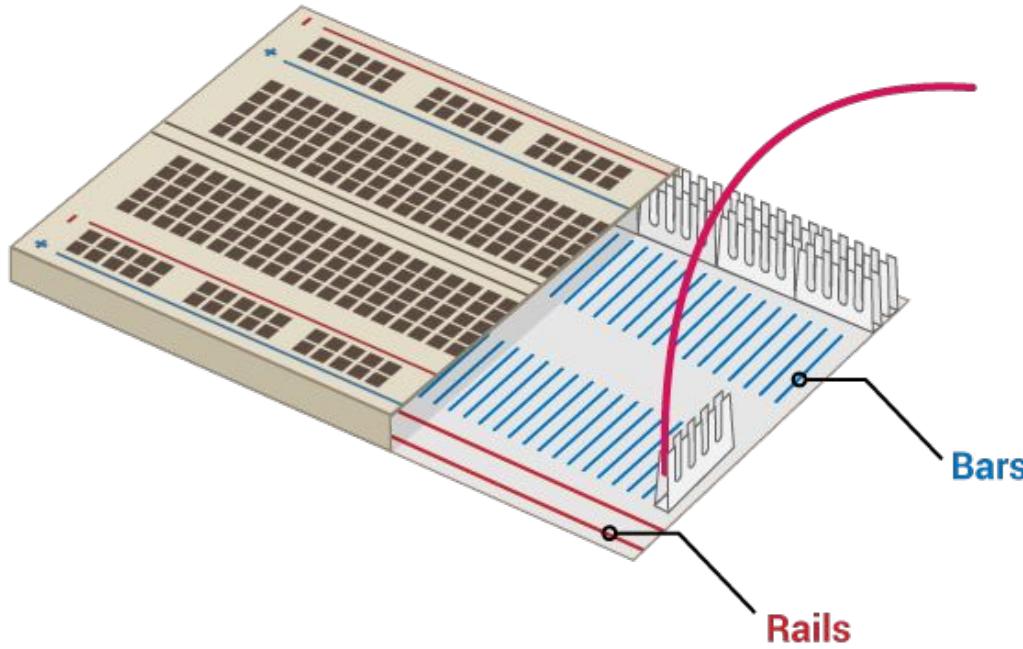
x 2



GPIO reference board

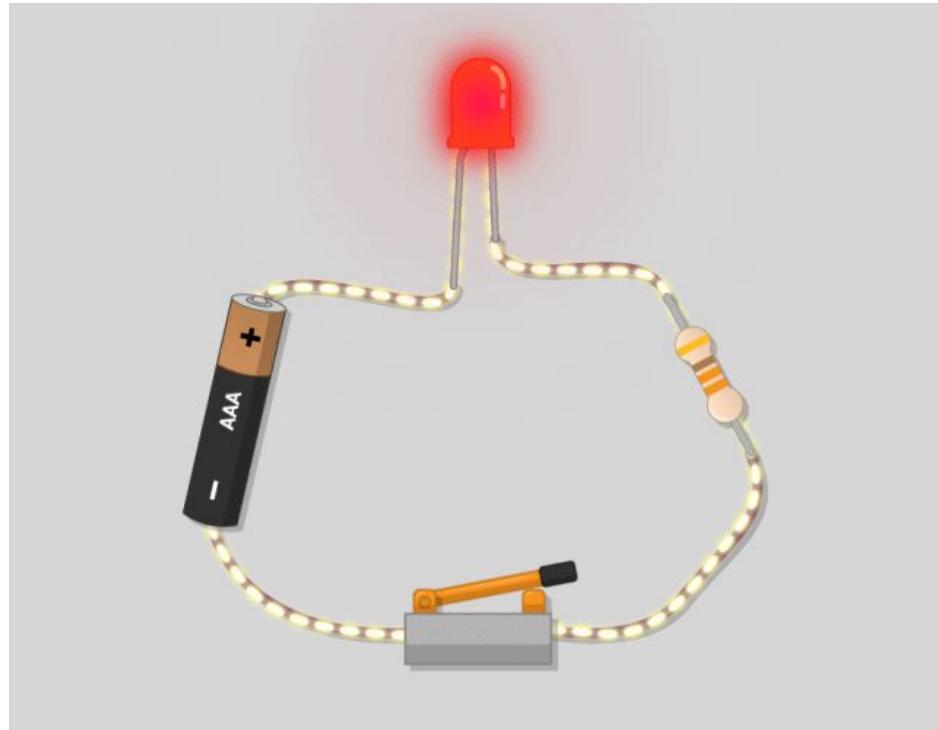


Inside a breadboard



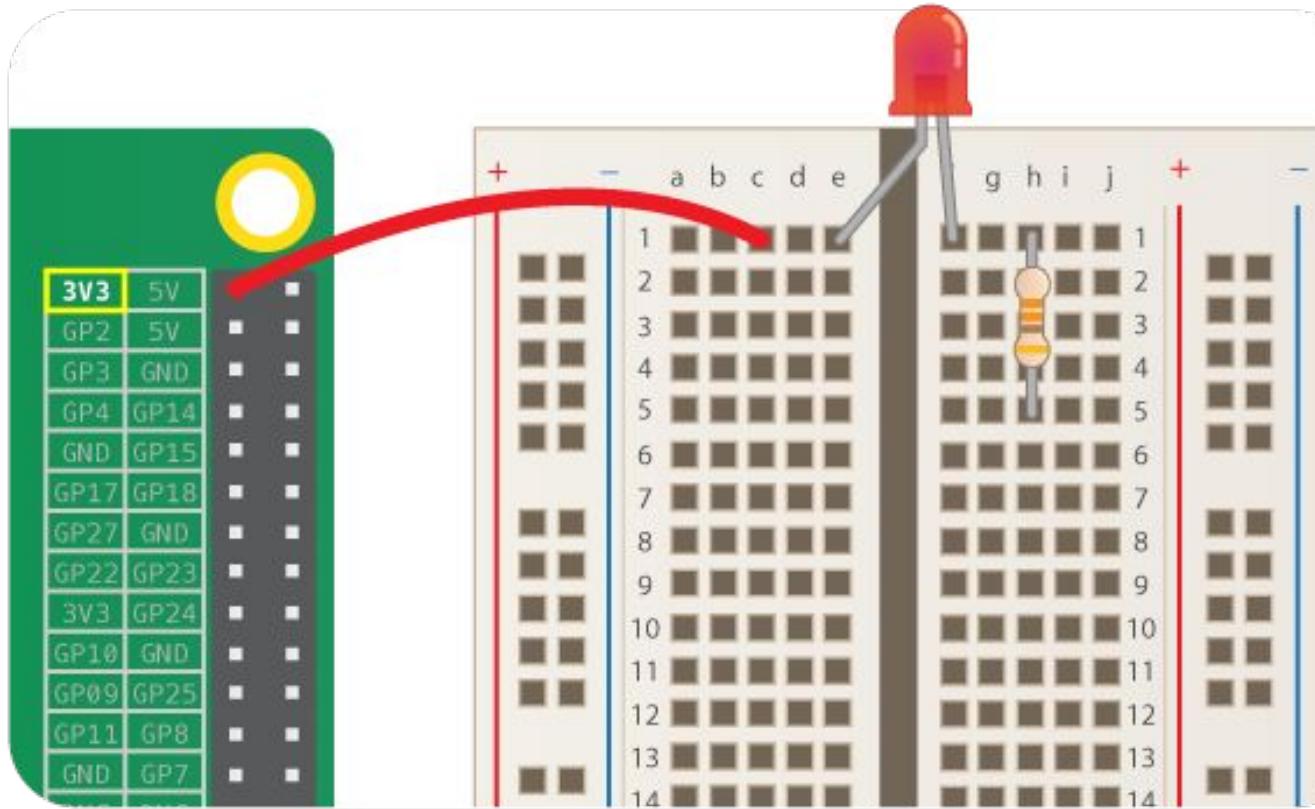
Simple Circuit

Your Raspberry Pi can act as the power supply for simple circuits.



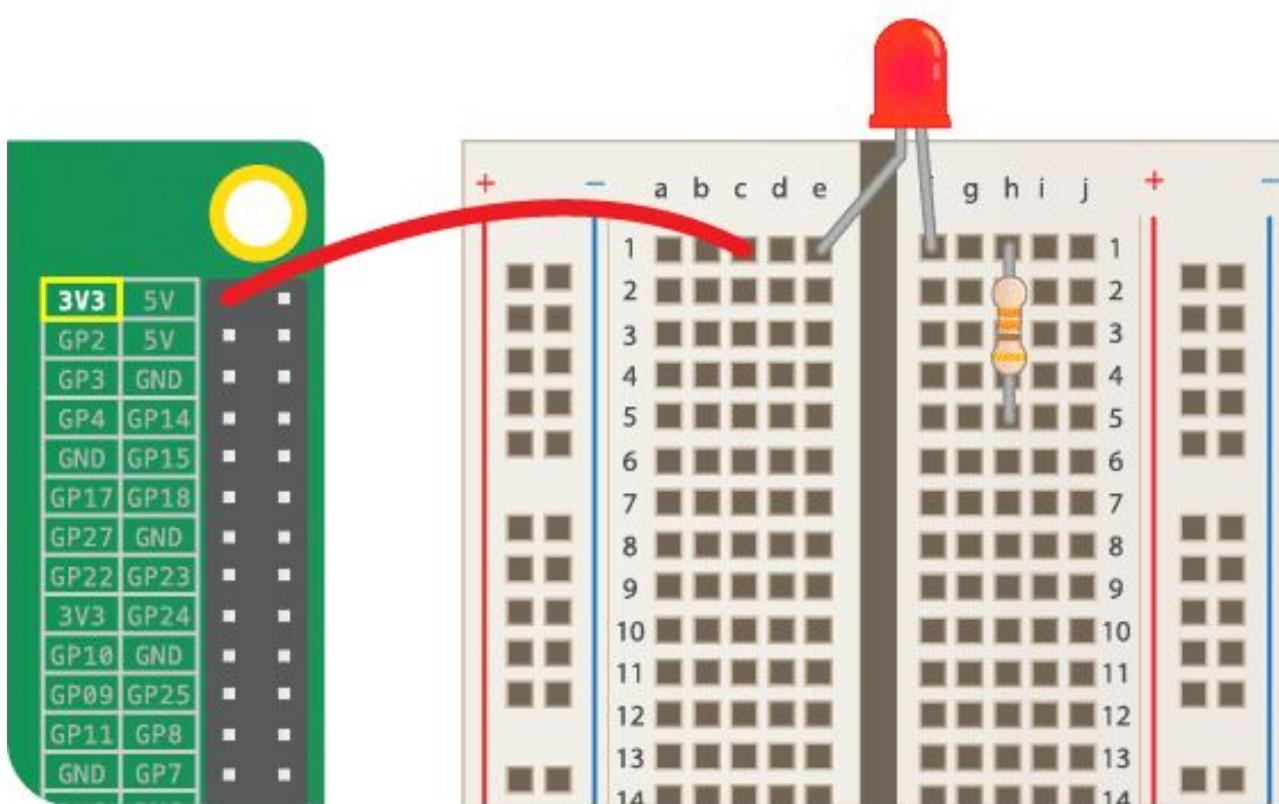
Simple Circuit

Let's build a simple circuit now using our Raspberry Pi.



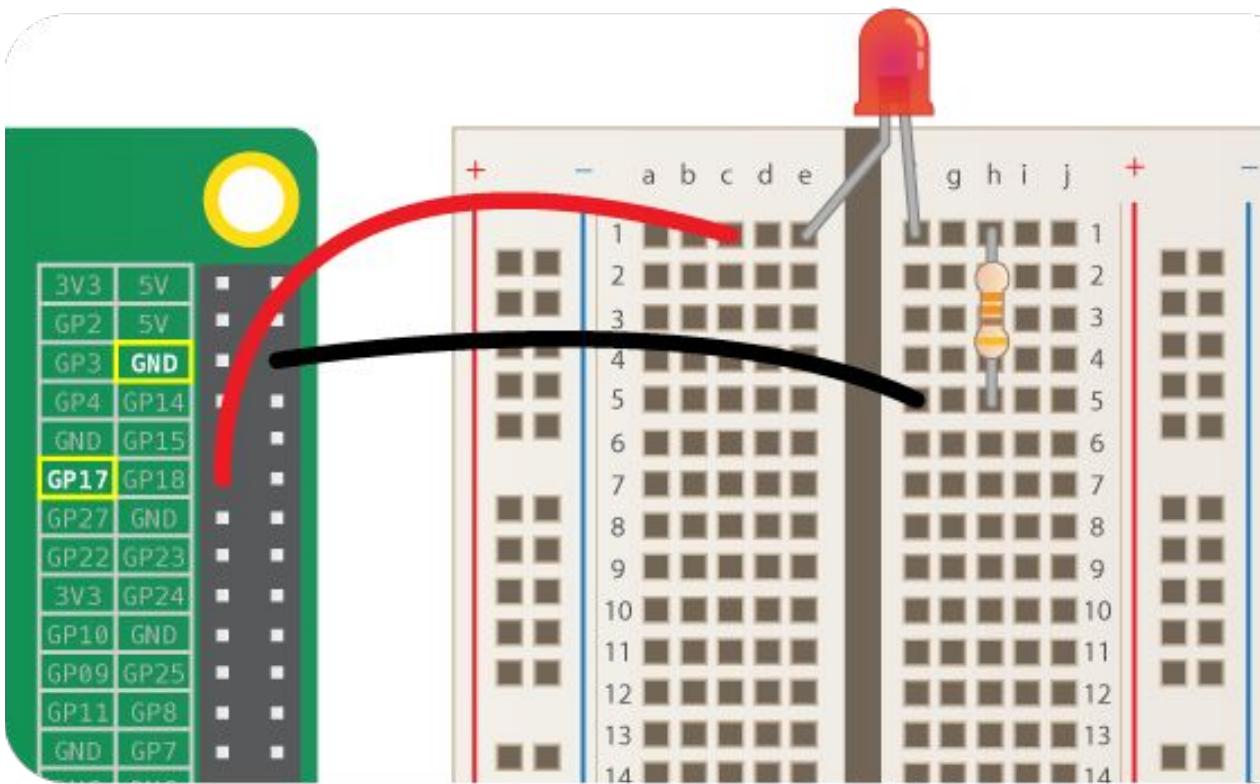
Simple Circuit

Now complete your circuit on the Raspberry Pi.



Switching Circuit

Your Raspberry Pi can act as a switch in the circuit, which can be controlled in software.



Scratch on the Raspberry Pi

Open Scratch and activate the GPIO server.



gpioserveron



Blinking LED

Setup your pin as an output.		config17out
Switch pin 17 on		gpio17on
Switch pin 17 off		gpio17off
Pause		
Loop forever		



Extra Challenge!

Can you:

- Flash your LED at different speeds, how fast can you make it flash?
- Can you make a dot (short flash) and dash (long flash) and use to make a distress beacon. S(...) O(---) S(...)



Python on the Raspberry Pi



Coding Your LED Light

```
# Flashing LED

from gpiozero import LED
from time import sleep

myled = LED(17)

while True:
    myled.on()
    sleep(1)
    myled.off()
    sleep(1)
```



Coding Your LED Light

```
# Flashing LED

from gpiozero import LED

led = LED(17)

led.blink()
```

Try some other values in led.blink(), what would these do:

- led.blink(5)
- led.blink(2, 0.5)
- led.blink(0.1, 10)
- led.blink(0.5, 0.5, 5, False)



Example Programs

```
# LED methods from the docs:
```

<https://gpiozero.readthedocs.io>

led.on() - Switches the Pin high

led.off() - Switches the Pin low

led.blink() - Makes the LED blink

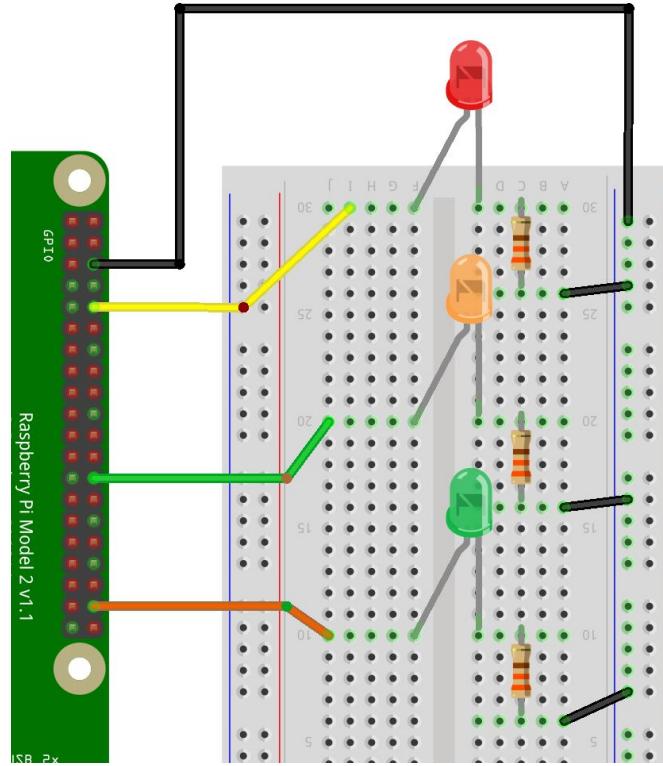
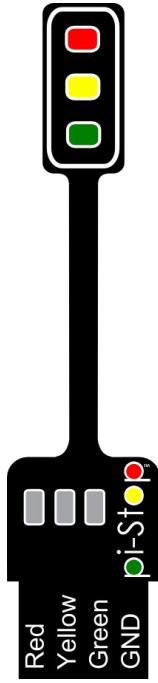
led.toggle() - Change the state of the LED

led.pin.number - Returns the pin number

led.is_lit - Returns the current state

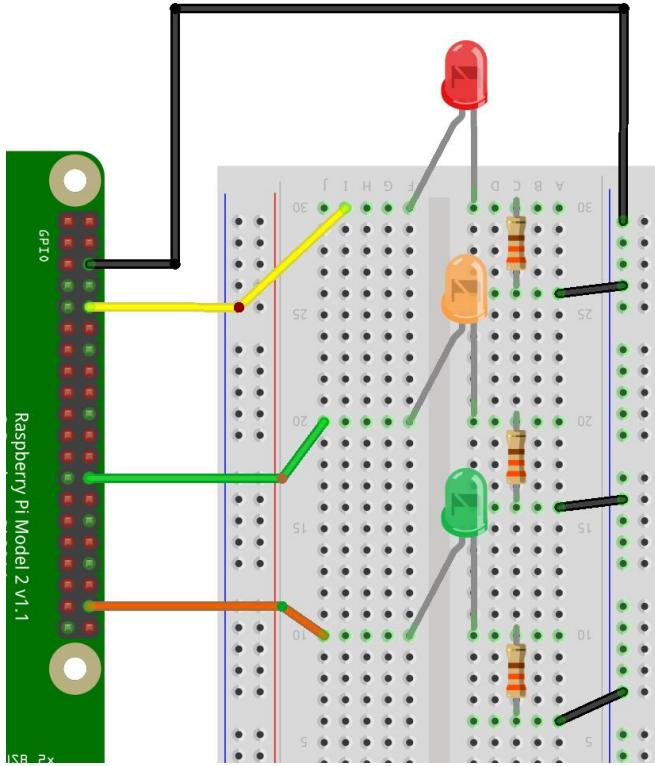


Hardware Application



Traffic Light

```
# Traffic Lights 1  
  
from gpiozero import LED  
from time import sleep  
  
red = LED(21)  
amber = LED(20)  
green = LED(16)  
  
red.on()  
sleep(3)  
red.off()  
amber.on()  
...  
...
```



Abstraction

Removing complexity to make a task more accessible

```
# Traffic Lights 1
```

```
from gpiozero import LED
from time import sleep

red = LED(21)
amber = LED(20)
green = LED(16)

red.on()
sleep(3)
red.off()
amber.on()

...
```

```
# Traffic Lights 2
```

```
from gpiozero import TrafficLights
from time import sleep

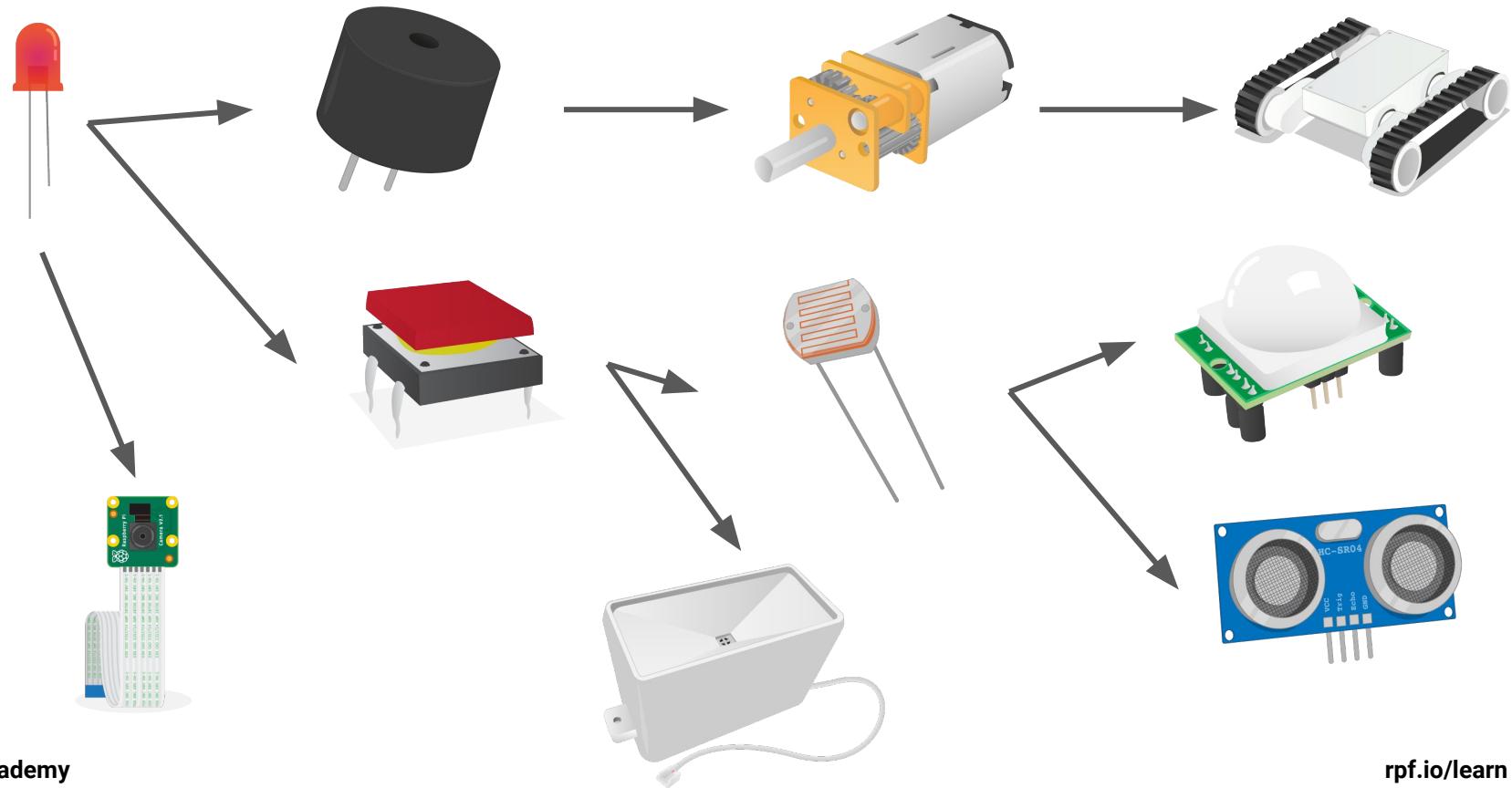
lights = TrafficLights(21, 20, 16)

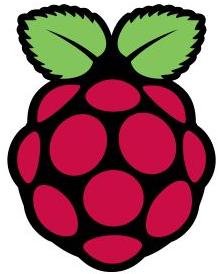
lights.red.on()
sleep(3)
lights.red.off()
lights.amber.on()

...
```



Where next?



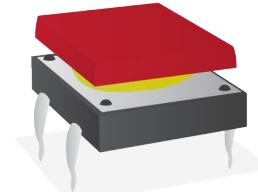
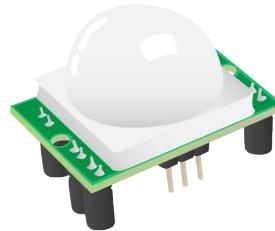


Pi Camera

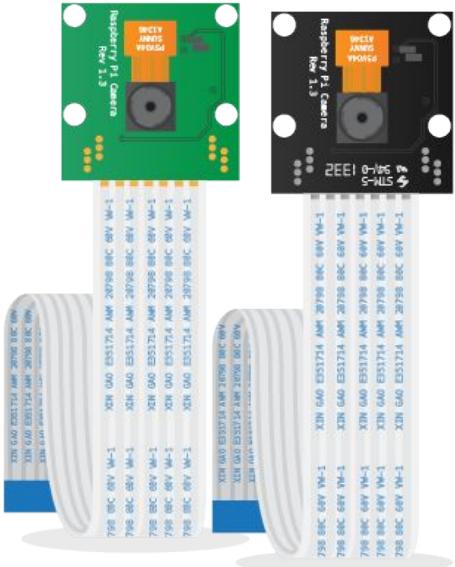
Taking Photos

Goals

- Connect the camera module
- Use Python to take pictures
- Add physical components to your project
- Use loops to repeat commands



Raspberry Pi camera module

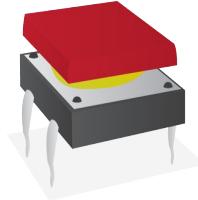
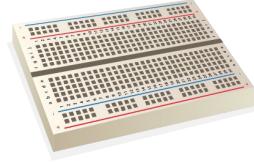
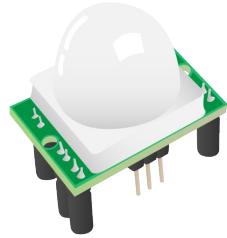


- 5Mpx / 8Mpx
- Full HD
- Photo & video
- Command line
- Python module
- Infra-red camera

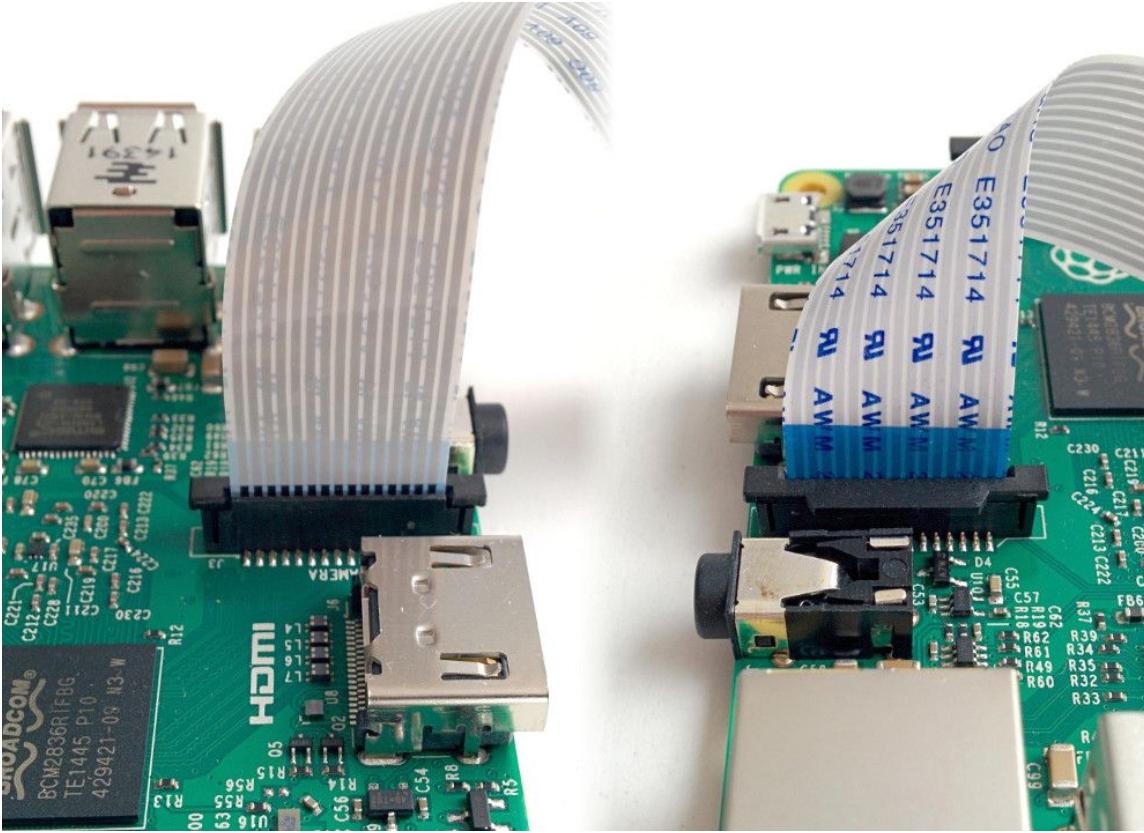


Materials

You will need:



Connect the camera



#picademy

rpf.io/learn



Camera Capabilities

Timelapse



High Speed recording



Sensor & Images



Infrared photography



Test your camera



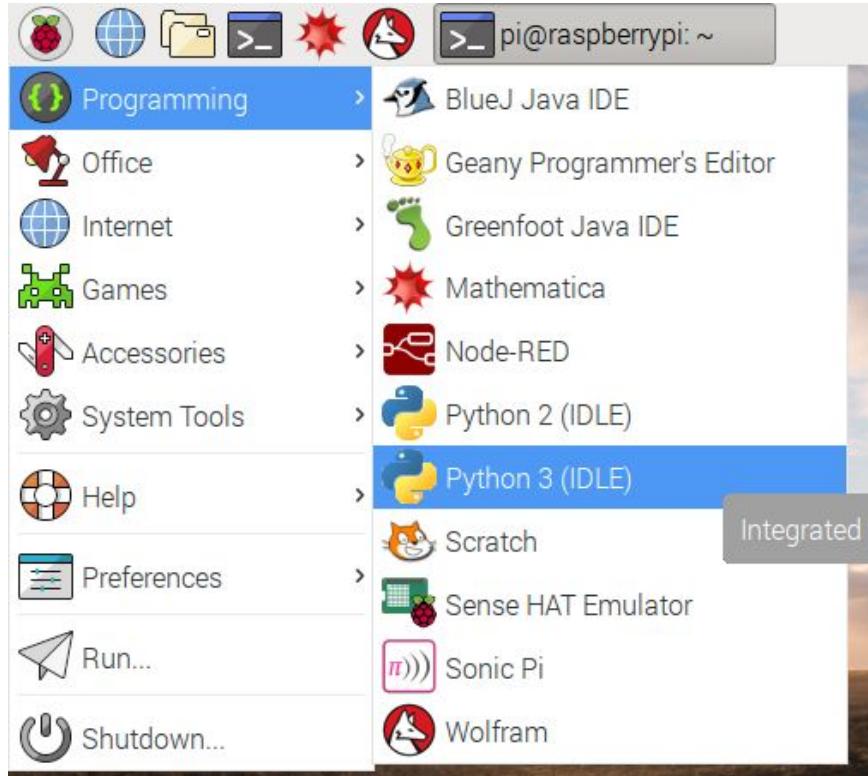
```
pi@raspberrypi:~ $ raspistill -k
```

[Ctrl + C](#) to close preview

```
pi@raspberrypi:~ $ raspistill -o image.jpg
```



Open Python 3 File



Take a selfie

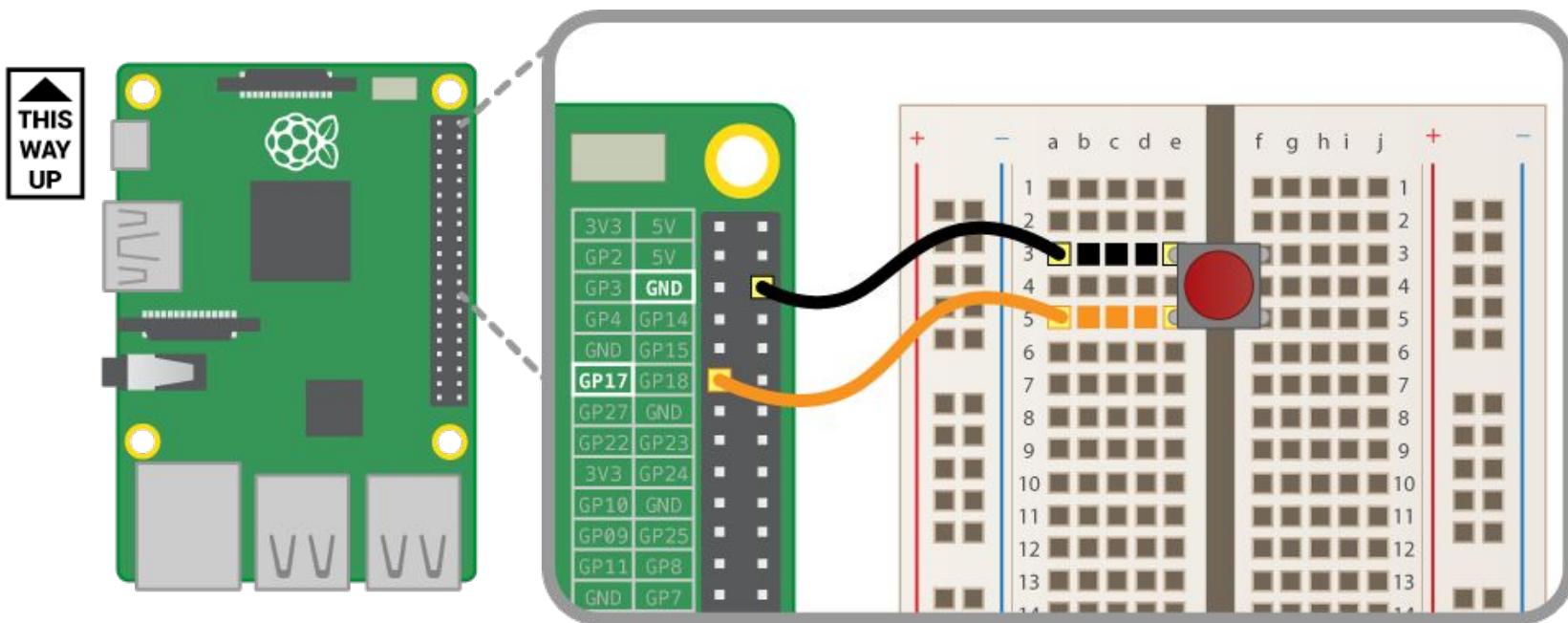
```
## selfie.py
from picamera import PiCamera
from time import sleep

camera = PiCamera()

camera.start_preview(alpha=192)
sleep(3)
camera.capture("/home/pi/image.jpg")
camera.stop_preview()
```



Add a Button



Add Button code

```
## button.py
from picamera import PiCamera
from gpiozero import Button
from time import sleep

camera = PiCamera()
button = Button(17)

camera.start_preview(alpha=192)
button.wait_for_press()
sleep(3)
camera.capture("/home/pi/button.jpg")
camera.stop_preview()
```



Add a loop

```
## loop.py
from picamera import PiCamera
from gpiozero import Button
from time import sleep

camera = PiCamera()
button = Button(17)

camera.start_preview(alpha=192)
for i in range(5):
    button.wait_for_press()
    sleep(1)
    camera.capture("/home/pi/button{0}.jpg".format(i))
camera.stop_preview()
```



What's the difference?

```
...
```

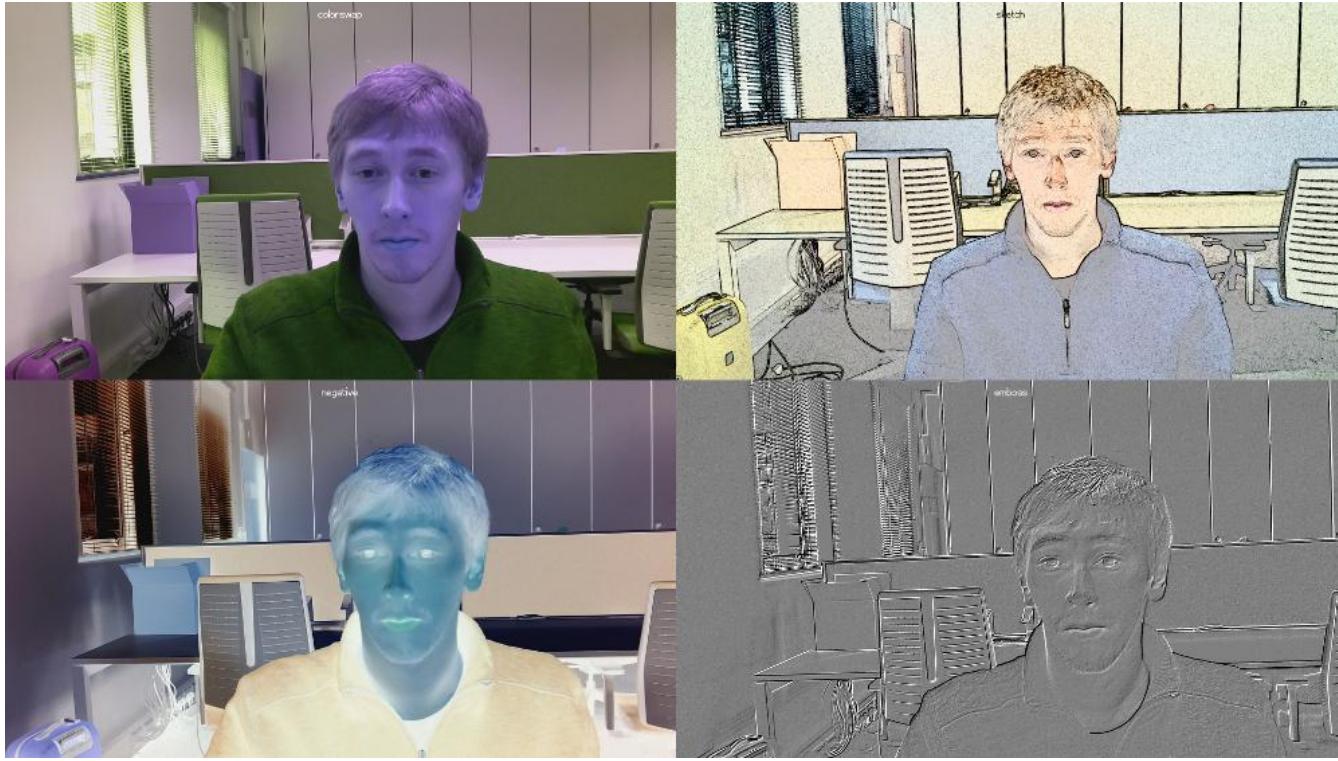
```
for i in range(5):
    button.wait_for_press()
    sleep(3)
    camera.capture("/home/pi/button{0}.jpg".format(i))
```

```
...
```

```
button.wait_for_press()
for i in range(5):
    sleep(3)
    camera.capture("/home/pi/button{0}.jpg".format(i))
```



Picamera effects



Picamera effects

```
...  
  
camera.start_preview(alpha=192)  
button.wait_for_press()  
camera.image_effect = 'negative'  
sleep(5)  
camera.capture("/home/pi/negative.jpg")  
camera.stop_preview()
```

1. Start with **selfie.py**
2. Save As **effect.py**

Try more effects:

- negative
- colorswap
- sketch
- emboss



Capturing video

```
##video.py
from picamera import PiCamera

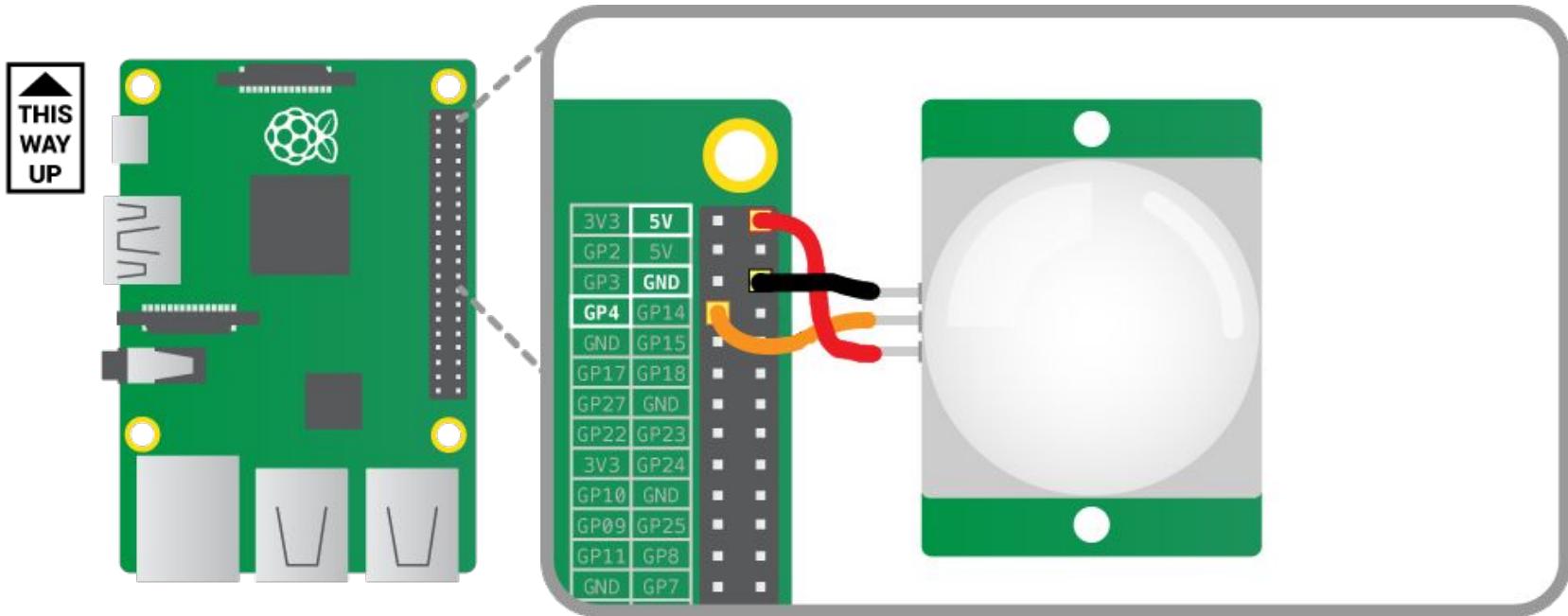
camera = PiCamera()
camera.start_preview(alpha=192)
camera.framerate = 24
camera.start_recording('my_video.h264')
camera.wait_recording(15)
camera.stop_recording()
camera.stop_preview()
```

Now play your video!

```
pi@raspberrypi:~ $  
omxplayer my_video.h264
```



Add a Motion Sensor



Add Motion Sensor code

```
## motion.py
from picamera import PiCamera
from gpiozero import MotionSensor
from time import sleep

camera = PiCamera()
sensor = MotionSensor(4)

camera.start_preview(alpha=192)
sensor.wait_for_motion()
sleep(3)
camera.capture("/home/pi/pir.jpg")
camera.stop_preview()
```



Button → Motion Sensor

```
# button.py
from picamera import PiCamera
from gpiozero import Button
from time import sleep

camera = PiCamera()
button = Button(17)

camera.start_preview(alpha=192)
button.wait_for_press()
sleep(3)
camera.capture("/home/pi/button.jpg")
camera.stop_preview()
```

```
## motion.py
from picamera import PiCamera
from gpiozero import MotionSensor
from time import sleep

camera = PiCamera()
sensor = MotionSensor(4)

camera.start_preview(alpha=192)
sensor.wait_for_motion()
sleep(3)
camera.capture("/home/pi/pir.jpg")
camera.stop_preview()
```



Capturing a time lapse

```
##timelapse.py
from picamera import PiCamera
from time import sleep

camera = PiCamera()
for num in range(1440):
    sleep(60)
    camera.start_preview(alpha=192)
    camera.capture("/home/pi/timelapse{0}.jpg".format(num))
    camera.stop_preview()
```



Documentation and help guides

picamera.readthedocs.io

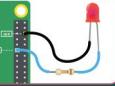
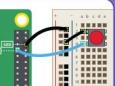
gpiozero.readthedocs.io

raspberrypi.org/resources

raspberrypi.org/education/downloads



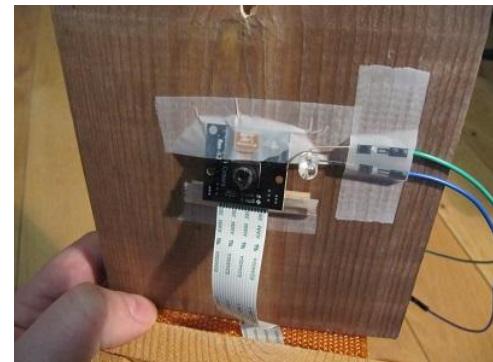
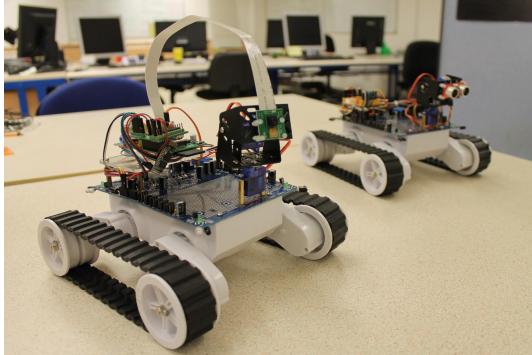
GPIO ZERO CHEATSHEET  raspberrypi.org/resources

LED  <pre>from gpiozero import LED led = LED(17) led.on()</pre> <table border="1"><tr><td>led.on()</td></tr><tr><td>led.off()</td></tr><tr><td>led.toggle()</td></tr><tr><td>led.blink()</td></tr></table>	led.on()	led.off()	led.toggle()	led.blink()	Full Colour LED  <pre>from gpiozero import RGBLED led = RGBLED(red=2, green=3, blue=4) r, g, b = 0, 0, 1 led.color = (r, g, b)</pre> <table border="1"><tr><td>led.on()</td></tr><tr><td>led.off()</td></tr><tr><td>led.color = (r, g, b)</td></tr><tr><td>led.red = 1</td></tr></table>	led.on()	led.off()	led.color = (r, g, b)	led.red = 1	Motor  <pre>from gpiozero import Motor motor = Motor(forward=17, backward=18) motor.forward()</pre> <table border="1"><tr><td>motor.forward()</td></tr><tr><td>motor.backward()</td></tr><tr><td>motor.stop()</td></tr><tr><td>motor.reverse()</td></tr></table>	motor.forward()	motor.backward()	motor.stop()	motor.reverse()	Button  <pre>from gpiozero import Button button = Button(4) while True: if button.is_pressed: print("Button is pressed") else: print("Button is not pressed")</pre> <table border="1"><tr><td>button.wait_for_press()</td></tr><tr><td>button.wait_for_release()</td></tr><tr><td>button.is_pressed</td></tr><tr><td>button.when_pressed = led.on</td></tr><tr><td>button.when_released = led.off</td></tr></table>	button.wait_for_press()	button.wait_for_release()	button.is_pressed	button.when_pressed = led.on	button.when_released = led.off	PIR Motion Sensor  <pre>from gpiozero import MotionSensor pir = MotionSensor(4) while True: if pir.motion_detected: print("You moved")</pre> <table border="1"><tr><td>pir.wait_for_motion()</td></tr><tr><td>pir.wait_for_no_motion()</td></tr><tr><td>pir.motion_detected</td></tr><tr><td>pir.when_motion = motor.forward</td></tr><tr><td>pir.when_no_motion = motor.backward</td></tr></table>	pir.wait_for_motion()	pir.wait_for_no_motion()	pir.motion_detected	pir.when_motion = motor.forward	pir.when_no_motion = motor.backward
led.on()																										
led.off()																										
led.toggle()																										
led.blink()																										
led.on()																										
led.off()																										
led.color = (r, g, b)																										
led.red = 1																										
motor.forward()																										
motor.backward()																										
motor.stop()																										
motor.reverse()																										
button.wait_for_press()																										
button.wait_for_release()																										
button.is_pressed																										
button.when_pressed = led.on																										
button.when_released = led.off																										
pir.wait_for_motion()																										
pir.wait_for_no_motion()																										
pir.motion_detected																										
pir.when_motion = motor.forward																										
pir.when_no_motion = motor.backward																										



What next?

- Time-lapse
- Stop motion animation
- Wildlife camera
- Send to social media
- Robotics



Reflection

- What do you think?
- How might you use the Raspberry Pi for projects, or in teaching?
- How can we improve the workshop?

Pi in the Sky



Pi in the Libraries: Lending

- 1 week+ loan
- Everything you need to get started
- 20-30 kits total
- Since 2013



Thanks!

More info: go.ncsu.edu/make
raspberrypi.org

Colin: crnickel@ncsu.edu

Adam: asroger2@ncsu.edu